Mat 2345

Week 5

Fall 2013

Student Responsibilities — Week 5

- **Reading**: Textbook, Section 2.4
- **Assignments**: See Assignment Sheet
- **Attendance**: Strongly Encouraged

Week 5 Overview

- 2.4 Sequences and Summations

2.4 Sequences, Summations, and Cardinality of Infinite Sets

- **Sequence**: a function from a subset of the natural numbers (usually of the form $\{0, 1, 2, \dots\}$ to a set S
- The sets
$$\{0, 1, 2, 3, \dots, k\}$$
and
$$\{1, 2, 3, \dots, k\}$$
are called **initial segments** of $\mathbb{N}$
- **Notation**: if f is a function from $\{0, 1, 2, \dots\}$ to S, we usually denote $f(i)$ by a_i and we write:
$$\{a_0, a_1, a_2, a_3, \dots\} = \{a_i\}_{i=0}^{k} \quad or \quad \{a_i\}_0^k$$
where k is the upper limit (usually ∞)

Sequence Examples

- Using **zero–origin** indexing, if $f(i) = \frac{1}{(i+1)}$, then the sequence
$$f = \{1, \tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \dots\} = \{a_0, a_1, a_2, a_3, \dots\}$$
- Using **one–origin** indexing, the sequence f becomes
$$f = \{\tfrac{1}{2}, \tfrac{1}{3}, \tfrac{1}{4}, \dots\} = \{a_1, a_2, a_3, \dots\}$$

Some Useful Sequences	
n^{th} Term	**First 10 Terms**
n^2	1, 4, 9, 16, 25, 36, 49, 64, 81, 100, ...
n^3	1, 8, 27, 64, 125, 216, 343, 512, 729, 1000, ...
n^4	1, 16, 81, 256, 625, 1296, 2401, 4096, 6561, 10000, ...
2^n	2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, ...
3^n	3, 9, 27, 81, 243, 729, 2187, 6561, 19683, 59049, ...
$n!$	1, 2, 6, 24, 120, 720, 5040, 40320, 362880, 3628800, ...

Summation Notation

Given a sequence $\{a_i\}_0^k$ we can add together a subset of the sequence by using the summation and function notation
$$a_{g(m)} + a_{g(m+1)} + \cdots + a_{g(n)} = \sum_{j=m}^{n} a_{g(j)}$$
or more generally
$$\sum_{j \in S} a_j$$

Examples

- $r^0 + r^1 + r^2 + r^3 + \cdots + r^n = \sum_{j=0}^{n} r^j$
- $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots = \sum_{i=1}^{\infty} \frac{1}{i}$
- $a_{2m} + a_{2(m+1)} + \cdots + a_{2(n)} = \sum_{j=m}^{n} a_{2j}$
- If $S = \{2, 5, 7, 10\}$, then $\sum_{j \in S} a_j = a_2 + a_5 + a_7 + a_{10}$

What are these sums?

- $\sum_{i=0}^{1} i^2 \ =$
- $\sum_{i=0}^{3} i^2 \ =$
- $\sum_{j=-1}^{1} 2^j \ =$
- $\sum_{k=3}^{5} (-1)^k \ =$

Product Notation

Similarly for multiplying together a subset of a sequence

$$\prod_{j=m}^{n} a_j \ = \ a_m a_{m+1} \dots a_n$$

Geometric Progression

Geometric Progression: a sequence of the form:

$$a, ar, ar^2, ar^3, ar^4, \dots$$

There's a proof in the textbook that

$$\sum_{i=0}^{n} r^i \ = \ \frac{r^{n+1}-1}{r-1} \text{ if } r \neq 1$$

You should be able to determine the sum:

- if $r \ = \ 0$
- if the index starts at k instead of 0
- if the index ends at something other than n (e.g., $n-1$, $n+1$, etc.)

Some Useful Summation Formulae	
Sum	**Closed Form**
$\sum_{k=0}^{n} ar^k, \ \ (r \neq 0)$	$\frac{ar^{n+1}-a}{r-1}, \ \ r \neq 1$
$\sum_{k=1}^{n} k$	$\frac{n(n+1)}{2}$
$\sum_{k=1}^{n} k^2$	$\frac{n(n+1)(2n+1)}{6}$
$\sum_{k=1}^{n} k^3$	$\frac{n^2(n+1)^2}{4}$
$\sum_{k=0}^{\infty} x^k, \ \ (\lvert x \rvert < 1)$	$\frac{1}{1-x}$
$\sum_{k=1}^{\infty} kx^{k-1}, \ \ (\lvert x \rvert < 1)$	$\frac{1}{(1-x)^2}$

Cardinality and Countability

The cardinality of a set A is **equal** to the cardinality of a set B, denoted $|A| \ = \ |B|$, if there exists a **bijection** from A to B.

A set is **countable** if it has the same cardinality as a subset of the natural numbers, $\mathbb{N}$

If $|A| \ = \ |\mathbb{N}|$, the set A is said to be **countably infinite**.

The (transfinite) cardinal number of the set $\mathbb{N}$ is

aleph null $\ = \ \aleph_0$

If a set is not countable, we say it is **uncountable**

Examples of Uncountable Sets

- The real numbers in the closed interval $[0, 1]$
- $\mathscr{P}(\mathbb{N})$, the power set of $\mathbb{N}$

Note: with infinite sets, **proper** subsets can have the same cardinality. This **cannot** happen with finite sets

Countability carries with it the implication that there is a **listing** or **enumeration** of the elements of the set

Definition: $|A| \leq |B|$ if there is an injection from A to B.

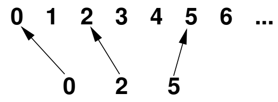
Theorem. If $|A| \leq |B|$ and $|B| \leq |A|$, then $|A| = |B|$.
This implies

- if there is an injection from A to B and
- if there is an injection from B to A

then

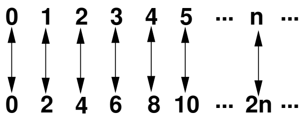
- there must be a bijection from A to B

- This is **difficult** to prove, but is an example of demonstrating existence without construction.
- It is often easier to build the injections and then conclude the bijection exists.
- Example I.
  Theorem: If A is a subset of B, then $|A| \leq |B|$.
  Proof: the function $f(x) = x$ is an injection from A to B
- Example II. $|\{0, 2, 5\}| \leq \aleph_0$
  The injection $f\{0, 2, 5\} \rightarrow \mathbb{N}$ defined by $f(x) = x$ is:



Some Countably Infinite Sets

- The set of even integers $\mathbb{E}$ is countably infinite...
  Note that $\mathbb{E}$ **is a proper subset of** $\mathbb{N}$
  Proof: Let $f(x) = 2x$. Then f is a bijection from $\mathbb{N}$ to $\mathbb{E}$

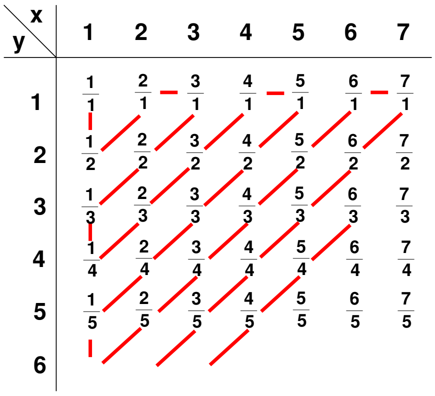


- $\mathbb{Z}^+$, the set of positive integers, is countably infinite
- The set of positive rational numbers, $\mathbb{Q}^+$, is countably infinite

---

Proof: $\mathbb{Q}^+$ is countably infinite

- $\mathbb{Z}^+$ is a subset of $\mathbb{Q}^+$, so $|\mathbb{Z}^+| = \aleph_0 \leq |\mathbb{Q}^+|$
- Next, we must show that $|\mathbb{Q}^+| \leq \aleph_0$.
- To do this, we show that the positive rational numbers with repetitions, $\mathbb{Q}_{\mathbb{R}}$, is countably infinite.
- Then, since $\mathbb{Q}^+$ is a subset of $\mathbb{Q}_{\mathbb{R}}$, it would follow that $|\mathbb{Q}^+| \leq \aleph_0$, and hence $|\mathbb{Q}^+| = \aleph_0$



- The position on the path (listing) indicates the image of the bijection function f from $\mathbb{N}$ to $\mathbb{Q}_{\mathbb{R}}$:
  $f(0) = \frac{1}{1}$, $f(1) = \frac{1}{2}$, $f(2) = \frac{2}{1}$, $f(3) = \frac{3}{1}$, etc.
- Every rational number appears on the list at least once, some many times (repetitions).
- Hence, $|\mathbb{N}| = |\mathbb{Q}_{\mathbb{R}}| = \aleph_0$

The set of all rational numbers, $\mathbb{Q}$, positive and negative, is also **countably infinite**.

More Examples of Countably Infinite

The set S of (finite length) strings over a finite alphabet A is countably infinite.

To show this, we assume that:

- A is non–empty
- There is an "alphabetical" ordering of the symbols in A

Proof: List the strings in lexicographic order —

- all the strings of zero length
- then all the strings of length 1 in alphabetical order,
- then all the strings of length 2 in alphabetical order,
- etc.

This implies a bijection from $\mathbb{N}$ to the list of strings and hence it is a countably infinite set

String Example

Let the alphabet $A = \{a, b, c\}$

Then the lexicographic ordering of the strings formed from A is:

$$\{\lambda, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, aaa, aab, aac, aba, \dots\}$$

$$= \{f(0), f(1), f(2), f(3), f(4), \dots\}$$

The Set of All C++ Programs is **countable**

Proof: Let S be the set of legitimate characters which can appear in a C++ program.

- A C++ compiler will determine if an input program is a syntactically correct C++ program (the program doesn't have to do anything useful).
- Use the lexicographic ordering of S and feed the strings into the compiler.
- If the compiler says YES, this is a syntactically correct C++ program, we add the program to the list.
- Else, we move on to the next string

In this way we construct a list or an implied bijection from $\mathbb{N}$ to the set of C++ programs.

Hence, the set of C++ programs is countable.

The Set of All Java Programs is **countable**

Proof: Let S be the set of legitimate characters which can appear in a Java program.

- A Java compiler will determine if an input program is a syntactically correct Java program (the program doesn't have to do anything useful).
- Use the lexicographic ordering of S and feed the strings into the compiler.
- If the compiler says YES, this is a syntactically correct Java program, we add the program to the list.
- Else, we move on to the next string

In this way we construct a list or an implied bijection from $\mathbb{N}$ to the set of Java programs.

Hence, the set of Java programs is countable.

Cantor Diagonalization

Cantor Diagonalization is an important technique used to construct an object which is **not** a member of a countable set of objects with (possibly) infinite descriptions

Theorem: The set of real numbers between 0 and 1 is **uncountable**.

Proof: We assume that it is countable and derive a **contradiction**.

Proof

- If the set is countable, we can list all the real numbers (i.e., there is a bijection from a subset of $\mathbb{N}$ to the set).
- We show that no matter what list you produce we can construct a real number between 0 and 1 which is not in the list.
- Hence, the number we constructed cannot exist in the list and therefore the set is not countable.
- It's actually much bigger than countable — it's said to have the **cardinality of the continuum**, $\mathfrak{c}$

Represent each real number in $(0,1)$ using its **decimal expansion**

E.g.	$\frac{1}{3}$	$=$	$0.3333333\ldots\ldots$
	$\frac{1}{2}$	$=$	$0.5000000\ldots\ldots$
		$=$	$0.4999999\ldots\ldots$

(It doesn't matter if there is more than one expansion for a number as long as our construction takes this into account.)

The resulting list:

$$\begin{aligned} r_1 &= 0.d_{11}d_{12}d_{13}d_{14}d_{15}d_{16}\ldots\ldots \\ r_2 &= 0.d_{21}d_{22}d_{23}d_{24}d_{25}d_{26}\ldots\ldots \\ r_3 &= 0.d_{31}d_{32}d_{33}d_{34}d_{35}d_{36}\ldots\ldots \\ &\vdots \end{aligned}$$

Now, **construct** the number $x = 0.x_1x_2x_3x_4x_5x_6x_7\ldots\ldots$ so that:

$$\begin{aligned} x_i &= 3 \text{ if } d_{ii} \neq 3 \\ x_i &= 4 \text{ if } d_{ii} = 3 \end{aligned}$$

Note: choosing 0 and 9 is not a good idea because of the non–uniqueness of decimal expansions.

Then, owing to the way it was constructed, x is **not equal** to any number in the list.

Hence, no such list can exist, and thus the interval $(0,1)$ is uncountable.

Computability

A number x between 0 and 1 is **computable** if there is a C++ (or Java, etc.) program which, when given the input i, will produce the i^{th} digit in the decimal expansion of x.

Example: The number $\frac{1}{3}$ is computable.

The C++ program which always outputs the digit 3, regardless of the input, computes the number

Some Things are Not Computable

Theorem. There exists a number x between 0 and 1 which is **not computable**.

There **does not exist** a C++ program (or a program in any other computer language) which will compute it!

Why? Because there are more numbers between 0 and 1 than there are C++ programs to compute them.

(In fact, there are $\mathfrak{c}$ such numbers!)

Yet another example of the non–existence of programs to compute things!